7Ш

УДК 539.3

Т.Б. Шатовская, А.В. Ляховец, И.В. Каменева
Харьковский национальный университет радиоэлектроники,
Украина, 61166, г. Харьков, проспект Ленина 14, irina.kamenieva@gmail.com

Модификация алгоритма построения графа в алгоритме Хамелеон

T.B. Shatovskaya, A.V. Lyahovec, I.V. Kameneva
Harkiv National University Radioelectronics
Ukraine, 61166, c. Kharkiv, ave. Lenina 14, irina.kamenieva@gmail.com

Modification of Algorithm for Graph Construction in the Chameleon Algorithm

Т.Б. Шатовська, А.В. Ляховець, І.В. Камєнєва
Харківський національний університет радіоелектроніки
Україна, 61166, м. Харків, проспект Леніна 14, irina.kamenieva@gmail.com

Модифікація алгоритму побудови графа в алгоритмі Хамелеон

В статье представлена модификация алгоритма Хамелеон. Алгоритм Хамелеон состоит из следующих этапов: построение графа, огрубление, разделение и восстановление. На каждом из этапов могут быть использованы различные подходы и алгоритмы. Рассмотрено 2 вида графов: симметричный *k-nn* граф и ассиметричный *k-nn* граф.
Ключевые слова: кластеризация, алгоритм Хамелеон, построение графа, связность, *k*-ближайших соседей.

In the article, modification of Chameleon algorithm is presented. Chameleon algorithm consists of the following stages: graph construction, coarsening, partitioning and uncoarsening. At each of these steps, different algorithms and approaches can be used. The main goal of this work is investigation and improvement of graph construction stage. This can be done by modification of *k*-selection algorithm during *k-nn* graph construction. It is considered two kinds of graphs: symmetric and asymmetric.
Key words: clustering, Chameleon algorithm, graph construction, connectivity, *k*-nearest neighbors.

У роботі представлений модифікований алгоритм Хамелеон. Алгоритм Хамелеон побудований з таких етапів: побудова графа, огрубіння, поділ та відновлення. На кожному з цих етапів можуть бути використані різні підходи та алгоритми. Головною метою роботи є дослідження з покращення этапу побудови через оптимізацію алгоритму вибору *k* під час побудови графа *k* найближчих сусідів. Розглянуто 2 види графів: симетричний *k-nn* граф та асиметричний *k-nn* граф.
Ключові слова: кластеризація, алгоритм Хамелеон, побудова графа, зв’язність, k-найближчих сусідів.

Введение

На данный момент весьма активно исследуются различные методы кластеризации. Каждым из целого множества имеющихся методов можно получить различные разбиения исходного множества. Выбор определенного метода зависит от типа желаемого результата. Производительность метода с определенными типами данных зависит от характеристик сервера и технических возможностей программного обеспечения, размера множества.

В последнее время ведутся активные разработки новых алгоритмов кластеризации, способных обрабатывать сверхбольшие базы данных. В них основное внимание уделяется масштабируемости. Разработаны алгоритмы, в которых методы иерархической кластеризации интегрированы с другими методами. К наиболее актуальным алгоритмам относятся: *BIRCH, CURE, CHAMELEON, ROCK* [1].

Главной целью работы является исследование и улучшение этапа построения графа посредством оптимизации алгоритма выбора *k* при построении графа *k* ближайших соседей.

Описание модифицированного алгоритма Хамелеон

Хамелеон – это новый иерархический алгоритм, который преодолевает ограничения существующих алгоритмов кластеризации. Данный алгоритм рассматривает динамическое моделирование в иерархической кластеризации [2-5].

В алгоритме можно выделить следующие этапы:

Хамелеон представляет объекты посредством часто используемого графа *k*-ближайших соседей (*k*-nearest neighbor graph). В данной работе рассмотрено 2 вида графов: симметричный *k-nn* граф и ассиметричный *k-nn* граф. При построении графа для каждой пары объектов измеряется «расстояние» между ними – степень похожести. Используются следующие меры: евклидово расстояние, квадрат евклидова расстояния, расстояние городских кварталов (манхэттенское расстояние), расстояние Минковского, расстояние Чебышева, степенное расстояние.

На следующем шаге строится очередь из последовательно уменьшенных гиперграфов – стадия огрубления (Coarsening Phase). Для огрубления графов может быть применено несколько существующих алгоритмов: случайное паросочетание, паросочетание из тяжелых ребер (HEM), модифицированный алгоритм паросочетания тяжелых ребер (Modified Heavy Edge Matching – HEM*), паросочетание из наиболее тяжелых ребер (heaviest-edge matching), модифицированное паросочетание из наиболее тяжелых ребер HEM*+, паросочетание легких ребер (Light Edge Matching (LEM)), паросочетание из тяжелых клик (HCM), сочетане тяжелых треугольников (Heavy-triangle matching (HTM)), сочетание тяжелых схем (Heaviest Schema Matching HSM), сочетание гиперребер (Hyperedge Coarsening HEC), видоизмененное сочетание гиперребер (Modified Hyperedge Coarsening MHEC), сочетание лучшего (первого) выбора (First Choice Coarsening FCC).

На третьей стадии выполняется разделение огрубленного графа таким образом, чтобы было удовлетворено ограничение баланса и оптимизирована функция разделения. Разделение может быть выполнено следующими методами: покоординатное разбиение (Coordinate Nested Dissection (CND)), деление сети с использованием кривых заполняющих пространство (Space-flling Curve Techniques), алгоритм возрастающего графа (GGP), алгоритм возрастающего графа с учетом выгод (GGGP), уровневое ячеечное разбиение (Levelized Nested Dissection – LND), Seed-Growth bisection, Kernighan-Lin Algorithm (KL), Fiduccia and Mattheyses.

На четвертом шаге выполняется восстановление графа. Разделение огрубленного графа проецируется на следующий уровень исходного графа и выполняется алгоритм улучшения разделения (partitioning refinement algorithm). Улучшение графа производится методами: Kernighan-Lin Algorithm (KL), Fiduccia and Mattheyses, граничный KL и граничный FM (Boundary KL and Boundary FM).

На последней итерации Хамелеона определяется показатель схожести между каждой парой кластеров [6]. На основании данной меры наиболее близкие кластеры объединяются.

В данной работе основное внимание будет уделено первому этапу – построению графа.

Определение *k* при построении *k-nn* графа

При решении поставленной задачи при построении графа *k* должно быть выбрано таким образом, чтобы соблюдалось условие связности построенного графа. Граф называется связным, если в нем для любых двух вершин имеется маршрут, соединяющий эти вершины. На практике применяется два принципиально различных порядка обхода, основанные на поиске в глубину и поиске в ширину соответственно.

Поиск в ширину. Вначале все вершины помечаются как новые. Первой посещается вершина a, она становится единственной открытой вершиной. В дальнейшем каждый очередной шаг начинается с выбора некоторой открытой вершины x. Эта вершина становится активной. Далее исследуются ребра, инцидентные активной вершине. Если такое ребро соединяет вершину x с новой вершиной y, то вершина y посещается и превращается в открытую. Когда все ребра, инцидентные активной вершине, исследованы, она перестает быть активной и становится закрытой. Если на данном этапе остались незакрытые вершины, то граф несвязный.

Поиск в глубину. Главное отличие от поиска в ширину состоит в том, что при поиске в глубину в качестве активной выбирается та из открытых вершин, которая была посещена последней. Основной алгоритм тот же, что и в случае поиска в ширину, только нужно очередь заменить стеком, а процедуру BFS – процедурой DFS.

Общая оценка трудоемкости такая же, как для поиска в ширину – $O(m+n)$

В каждом из алгоритмов есть возможность оперировать не вершинами, а ребрами, связывающими вершины. Существует два варианта алгоритма поиска в глубину. Поиск в глубину с вычислением глубинных номеров – рекурсивный и итеративный варианты. Но так как в данном случае вычисление глубинных номеров не является необходимым, эти алгоритмы не являются актуальными. Приведенные алгоритмы поиска в глубину и ширину так же могут быть реализованы с использованием рекурсии, но в данном случае есть несколько ограничений и недостатков [7].

В худшем случае (при полном графе) рекурсивный алгоритм, перебирая все возможные ребра, будет вынужден вызвать основную процедуру $(N-1)!$ раз. Велика вероятность, что при достаточно большом N произойдет переполнение оперативной памяти, которое вызовет ошибку. Кроме того, размеры квадратной матрицы смежности дают сильное ограничение на возможное количество вершин графа: не более 250.

Итеративный же алгоритм переберет все ребра графа, которых может быть не более чем $N*(N+1)/2$. Следовательно, общая сложность алгоритма может быть приблизительно оценена значением $N^3/8$. Возможное количество вершин графа ограничено только максимальным размером линейного массива (32 000).

Таким образом, значение *k* последовательно увеличивается, пока граф не станет связным. Так как данная операция трудоемка и длительна, она нуждается в оптимизации.

Создание экспериментальных выборок

Для проверки работоспособности метода необходимо большое количество выборок. Отсутствие реального источника данных требуемого объема, разнообразия и

качества вынуждает обратиться к альтернативному источнику. Так как при использовании различных входных данных с определенными статистическими характеристиками производительность и качество кластеризации может сильно отличаться, необходимо проводить анализ на синтетических выборках, созданных специально для данной задачи. Исследований в данной области немного, и все крайне специфичны для рассматриваемых задач.

Существует ряд методов генерации экспериментальных данных, позволяющих провести анализ кластеризации систематически и последовательно. Такие генераторы используют параметризованные модели, которые создают реалистичные данные. Эти генераторы обучены на реальных данных.

Helmers и Bunke (2003) работали с образцами почерка. Baird (2000) и Baird (1993) работали с изображениями. Rogers и др. (2003) работали с 2D изображениями белка. Davidov и соавторы (2004) получали наборы данных, помечая текстовый контент из WWW. (Srikant, 1999). GSTD (Theodoridis и соавторы, 1999) и Jeske и соавторы (2005) также занимались синтетическим созданием данных. GSTD моделирует броуновское движение. Существует ряд скрытых моделей Маркова (Hidden Markov Model – HMM) на основе генераторов данных. Rachkovskij и Kussul (1998) продемонстрировали более общий алгоритм генерации образцов из признаков в пространстве, включая фоновый шум. Pei и Zaiane (2006) занимались получением данных для неконтролируемого обучения и обнаружения выбросов. Van der Walt и Bernard (2007) демонстрируют полезность синтетических генераторов набора данных на основе различных плотностей.

В статье Vineet Chaoji, Mohammad Al Hasan, Saeed Salem и Mohammed J. Zaki «SPARCL: Efficient and Effective Shape-based Clustering» для тестов масштабируемости, а также для создания 3D-данных написан собственный генератор кластеров, основанный на фигурах. Для создания фигуры в 2D случайным образом выбирались точки на канве и добавлялись точки, которые формируют желаемую фигуру. Точкой отсчета для всех фигур являлась точка (0,0).

Чтобы получить сложные фигуры, использовались фигуры, полученные вращением и смещением (круг, прямоугольник, эллипс, круговые полосы и т.д.). Генерация 3d фигуры построена на 2d фигуры. Случайным образом выбираются точки для третьей координаты – если координаты x и y удовлетворяют фигуры, случайным образом выбирается z-ось в пределах заданного диапазона.

Такой подход позволяет построить правдоподобные 3d фигуры, а не только несколько слоев 2d фигуры. Как и в случае 2d, комбинируется вращение и смещение 3d фигуры, чтобы получить более сложные фигуры – пример синтетических 3d данных. Как только созданы все фигуры, случайным образом добавляется шум (от 1% до 2%). Показанный на рисунке 3d набор данных имеет 100 000 точек, и 10 кластеров.

В данной работе создание 3D фигур выполняется посредством 3d s max studio. Данное приложение позволяет сгенерировать трехмерную фигуру необходимой плотности и с необходимым количеством точек. Далее фигура может быть экспортирована. Статистические характеристики полученной выборки будут зависеть от характера фигур, их размера, плотности и расположения. Данные параметры подбираются при создании фигур. Добавление шума в выборку производится непосредственно перед проведением анализа.

Для проведения эксперимента данным методом было сгенерированно 50 выборок с различными статистическими характеристиками.

Возможно, не все выборки одинаково хороши для изучения и сравнения алгоритмов кластеризации. Некоторые могут не иметь хорошо выраженную структуру кластеров, кластеры могут быть неоднородны, распределены различным образом. Все эти факторы влияют на качество работы алгоритмов.

Сгенерированные выборки – это достаточно простой и точный способ проведения эксперимента на большом количестве выборок с известными структурными характеристиками. Среди недостатков можно перечислить:

– Зависимость сгенерированных выборок от программы генератора. Плотность и количество вершин может быть различным даже при одинаковых параметрах для генерации выборки.

– Данный способ создания выборки позволяет создать набор с определенными характеристиками, но такие выборки не всегда могут иметь такую структуру. В некоторых предметных областях все еще тяжело воссоздать структуру реальных данных при генерации выборки.

Большое количество реальных выборок использовано в работе. Список ресурсов с реальными выборками представлен в табл. 1. Многие выборки не подходят для исследования в рамках данной работы количеством атрибутов или иными характеристиками. Для некоторых множеств данных могут быть использованы не только множества, но и подмножества.

Таблица 1 – Ссылки на ресурсы с наборами данных для кластеризации

People	http://people.sc.fsu.edu/~jburkardt/datasets/datasets.html
Weka	http://weka.wikispaces.com/Datasets
Cologne University	http://www.uni-koeln.de/themen/statistik/data/index.e.html
Standard datasets	http://cs.joensuu.fi/sipu/datasets/
D Star	http://uisacad2.uis.edu/dstar/data/clusteringdata.html
UCI KDD	http://kdd.ics.uci.edu/
UCI	http://archive.ics.uci.edu/ml/
Cha learn	http://www.causality.inf.ethz.ch/repository.php

Построение математической модели

Для оптимизации выбора начального параметра *k* при построении *k-nn* графа необходимо построить математическую модель зависимости *k* от характеристик обрабатываемой выборки. Математическая модель будет построена на основе исследования 30 выборок.

Математической моделью называется формальная система, которая представляет собой конечное собрание символов и совершенно точных правил оперирования с этими символами в совокупности с интерпретацией свойств определенного объекта некоторыми символами, отношениями и константами

Будем считать, что зависимости между параметрами задаются в виде следующего набора функций: $W_i = F(X_1, X_2, ..., X_n, a_1, a_2, ..., a_k), i = (1, m)$, где W – обозначения целевых параметров, X – обозначения управляемых параметров, a – обозначения неуправляемых параметров, m – число целевых параметров, n – число управляемых параметров, k – число неуправляемых параметров.

В данной работе управляемыми параметрами являются характеристики выборки, такие, как количество объектов в выборке, минимальные и максимальные значения матожидания, дисперсии и разброса.

В результате была получена следующая матмодель:

*Y = (1.06958e+008 - 605201*x1 - 7121.75*x2 + 5.97909*x3 - 4.69129e-005*x4) / (2.89754e+007 - 234795*x5 - 1118.79*x6 + x7.*

На рис. 1 представлено двумерное отображение описания данных матмоделью.

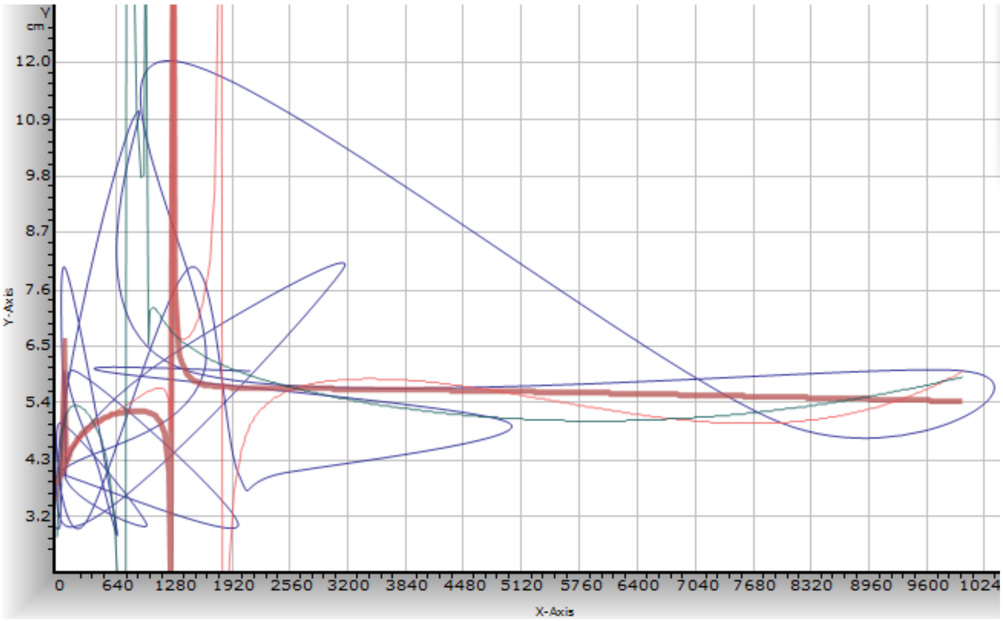


Рисунок 1 – Описание данных математической моделью

Анализ результатов

Ключевые аспекты оценивания – это эффективность, надежность, простота и результативность. Расчет времени производился на 1.73 GHz Intel (R) Pentium (R) Dual CPU с 2GB памяти.

Время производительности для разных графов может быть использовано для изучения масштабируемости алгоритма. Производительность зависит от количества вершин и ребер.

Использование данной модели при построении графа в рамках модифицированного алгоритма Хамелеон позволило сократить время построения графа до 16%. Использование модели особенно критично для больших выборок и больших значений матожидания и разброса.

Литература

1. Чубукова И.А. Data Mining БИНОМ / А.И. Чубукова // Лаборатория знаний. Интернет-университет информационных технологий. – ИНТУИТ.ру. – 2008.
2. Karypis G. Multilevel k-way Partitioning Scheme for Irregular Graphs / G. Karypis, V. Kumar // Journal of parallel and distributed computing. – 1998. – № 48. – P. 96-129
3. Karypis G. A fast and highly quality multilevel scheme for partitioning irregular graphs / G. Karypis, V. Kumar // SIAM J. Sci. Comput., to appear. [Also available on WWW at URL http://www.cs.umn.edu/_karypis]. – 1995.
4. Brian Read Advances in Databases : 18th British National Conference on Databases, BNCOD 18 Chilton, UK, July 9 – 11. – 2001.
5. Karypis G. Multilevel k-way Partitioning Scheme for Irregular Graphs / G. Karypis, V. Kumar // Society of Industrial and Applied mathematics. – 1999.
6. Karypis G. Chameleon: Hierarchical Clustering Using Dynamic Modeling / G. Karypis, E.-H. (Sam) Han, V. Kumar // Computer. – 1999. – Vol. 32, № 8. – P. 68-75.

Literatura

1. Chubukova I.A. Data Mining BINOM. Laboratorija znanij, Internet-universitet informacionnyh tehnologij. INTUIT.ru. 2008
2. Karypis G. Journal of parallel and distributed computing. 1998. № 48. P. 96-129.

3. Karypis G. and Kumar V. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput., to appear. [Also available on WWW at URL http://www.cs.umn.edu/_karypis]. 1995.
4. Brian Read Advances in Databases: 18th British National Conference on Databases, BNCOD 18 Chilton, UK, July 9-11. 2001.
5. Karypis G. and Kumar V. Multilevel k-way Partitioning Scheme for Irregular Graphs // Society of Industrial and Applied mathematics. 1999.
6. Karypis G. Computer. Vol. 32. 1999. № 8. P. 68-75.

RESUME

T.B. Shatovskaya, A.V. Lyahovec, I.V. Kameneva

Modification of Algorithm for Graph Construction in the Chameleon Algorithm

Today, very actively investigate various clustering methods.

Recently, new clustering algorithms that can handle extremely large databases are actively developing.

In the article modification of Chameleon algorithm is presented. The Chameleon algorithm consists of next stages: graph build, coarsening, partitioning and uncoarsing. On each of these stages different algorithms and approaches can be used.

The main goal of this work is investigation and improvement graph build stage. This can be done by modification of k selection algorithm during K-NN (K-Nearest Neighbors) graph building.

We consider symmetric K-NN graph and asymmetric K-NN graph.

During construction of the graph for each pair of objects measured the "distance" between them named as similarity. We can use the following measures: Euclidean distance, squared Euclidean distance, city block distance (Manhattan distance), Minkowski distance, Chebyshev distance. The next step is constraction of all successively reduced hypergraphs named as Coarsening Phase.

In the third step performs the division of coarsened graph that separation function was optimized and limit balance was satisfied.

The fourth step is graph recovery.

The separation of coarsened graph is project to the next level of the original graph and the partitioning refinement algorithm is executed

Статья поступила в редакцию 01.06.2012.